EXHIBIT I

Exhibit 1 – MANTRA I

’156 Patent

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156a**] A device comprising:	MANTRA I discloses a device. Specifically, the MANTRA I is a neurocomputer. *See, e.g.*: “The MANTRA I computer is a massively parallel machine dedicated to neural-network algorithms (Fig. 1). It has been designed to provide the basic operations for the following models: (1) single-layer networks (Perceptron and delta rule); (2) multilayer feedforward networks (back-propagation rule); (3) fully connected recurrent networks (Hopfield model); and (4) self-organizing feature maps (SOFMS; Kohonen model). A description of these algorithms can be found in any classic introductory book on neural networks (e.g., [4]). The Kohonen feature maps are used in Section 3 to illustrate how these algorithms are mapped on the system. The MANTRA I accelerator is based on a bidimensional systolic array composed of custom PEs named GENES IV. In the present section, the hardware of the machine is overviewed starting from its system integration in a network of workstations and proceeding down to the internal architecture of the machine and of its computational core.” Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 48.

Exhibit 1 – MANTRA I

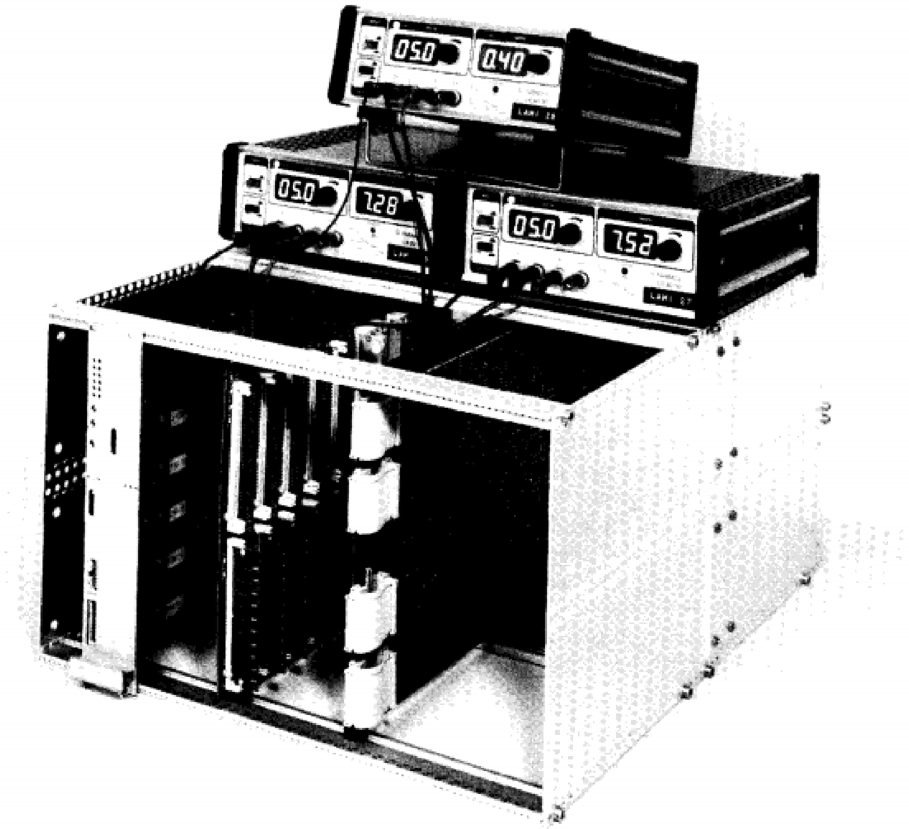
Claim Limitation (Claim 7)	Exemplary Disclosure
	 **FIGURE 1** The MANTRA I system. Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 48. "The machine, which is currently under test, consists of four printed circuit boards : (1) the processor board (34 chips), (2) the control board (135 chips), (3) the input/output board (465 chips including 14 GACD1 chips), (4) the GENES IV array board (containing a matrix of 10

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	x10 = 100 GENES IV chips). One of the latter board is required for configurations of the machine up to 20 x 20 = 400 PEs. For larger configurations—up to 40 x 40 = 1600 PEs—four such boards should be interconnected. A more detailed description of the MANTRA I machine can be found in [vir93].” Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 3.

Exhibit 1 – MANTRA I

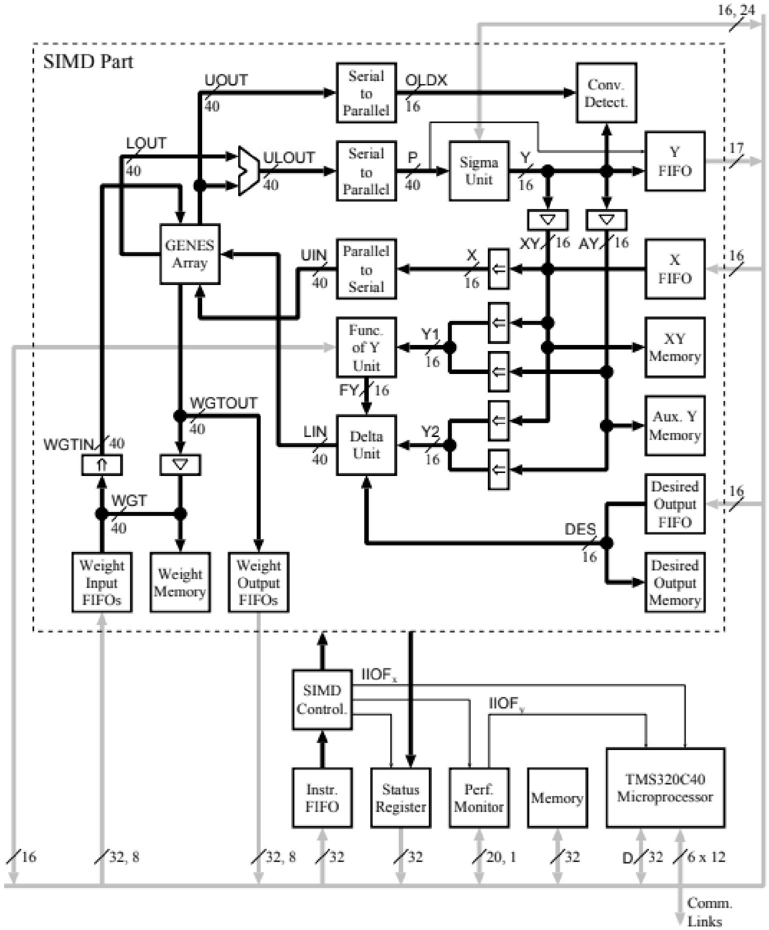
<table>
<tr><th>Claim Limitation (Claim 7)</th><th>Exemplary Disclosure</th></tr>
<tr><td></td><td>


Figure 4: Architecture of the MANTRA I machine.

Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf.
</td></tr>
</table>

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156b**] at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	MANTRA I discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See, e.g.*: “The systolic array at the heart of the SIMD part of the machine is a square mesh of GENES IV PEs [9], each connected by serial lines to its four neighbours, as shown in Figure 4. All input and output operations are performed by the PEs located on the north-west to south-east diagonal.” Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 50. “The architecture of the implemented array and the structure of the PEs are illustrated in figure 2. Unlike the basic architecture shown in figure 1 (a), the access to the processing array is done through input and output ports located on the top-left to bottom-right diagonal. These inputs (Uin and Lin) and outputs (Uout and Lout) are present on all PEs for modularity but are used only on diagonal cells. They are always bypassed or left unconnected on non-diagonal cells. As figure 2 (b) shows, two further registers (U and L) are required for this path. The advantage is that no external hardware is required to diagonalize the inputs or un-diagonalize the outputs to and from the array (see figure 1 (a)). All the components of a vector are entered or read at the same time. The transposition mechanism, described by equations (15) and (16), is added to the PE by providing two control signals from the systolic instruction unit. A first signal exchanges the roles of Nin and Win, while the other exchanges Sout and Eout, as shown in figure 3. When a computation on the transpose matrix begins, the inputs are exchanged. The quantities in the PE registers, belonging to the previous computation are output on the regular path. On the next computation step, the outputs are also exchanged. The same applies when the operation mode reverts to the direct matrix. ” Paolo Ienne & Marc A. Viredaz, *GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator*, Proceedings of the International Conference on Application-Specific Array Processors, Venice, Italy, October 1993, at 350-51.

Exhibit 1 – MANTRA I

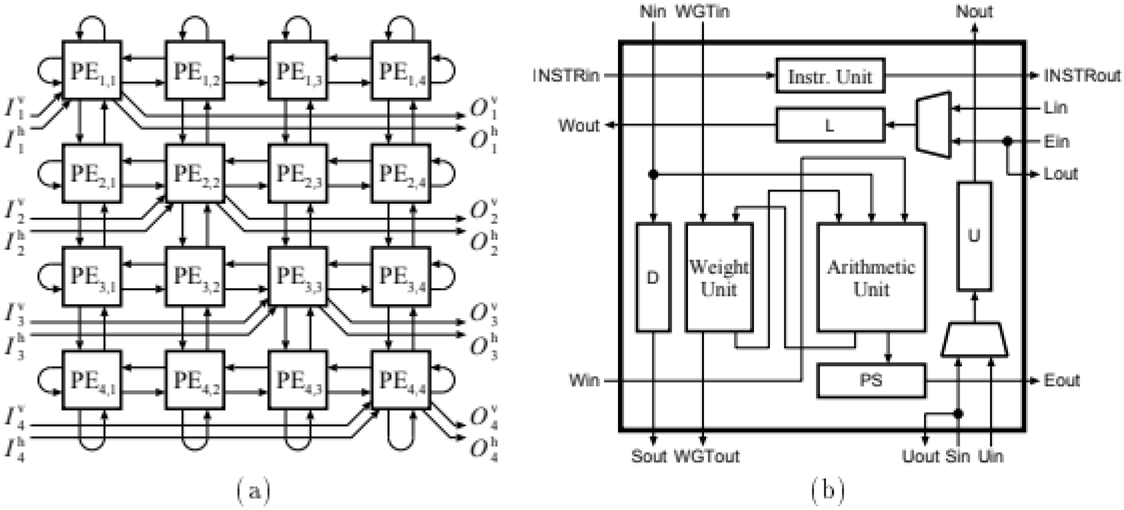
Claim Limitation (Claim 7)	Exemplary Disclosure
	 PE1,1 PE1,2 PE1,3 PE1,4 PE2,1 PE2,2 PE2,3 PE2,4 PE3,1 PE3,2 PE3,3 PE3,4 PE4,1 PE4,2 PE4,3 PE4,4 I1v I1h I2v I2h I3v I3h I4v I4h O1v O1h O2v O2h O3v O3h O4v O4h (a) Nin WGTin Nout INSTRin Instr. Unit INSTRout Wout L Lin Ein Lout D Weight Unit Arithmetic Unit U Win PS Eout Sout WGTout Uout Sin Uin (b) **Figure 2. (a) 4 × 4 array architecture. (b) Processing element basic structure.** Paolo Ienne & Marc A. Viredaz, *GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator*, Proceedings of the International Conference on Application-Specific Array Processors, Venice, Italy, October 1993, at 350-51. "The GENES IV array is a square mesh of dedicated processing elements (PEs). Each PE holds one element of the synaptic weight matrix. It is connected by serial lines to its four neighbors as shown in figure 1(a). All input and output operations take place on the north-west to south-east diagonal. The GENES IV array implements six operations: **Matrix-vector product**: this operation is an implementation of the classic systolic multiplier. It can alternatively be viewed as the scalar or dot product between a vector and each row of the matrix. **Squared Euclidean distance**: using the same data flow as the matrix-vector product and a slightly modified arithmetic unit, this operation computes the squared Euclidean distance between a vector and each row of the synaptic weight matrix stored in the array.

Exhibit 1 – MANTRA I

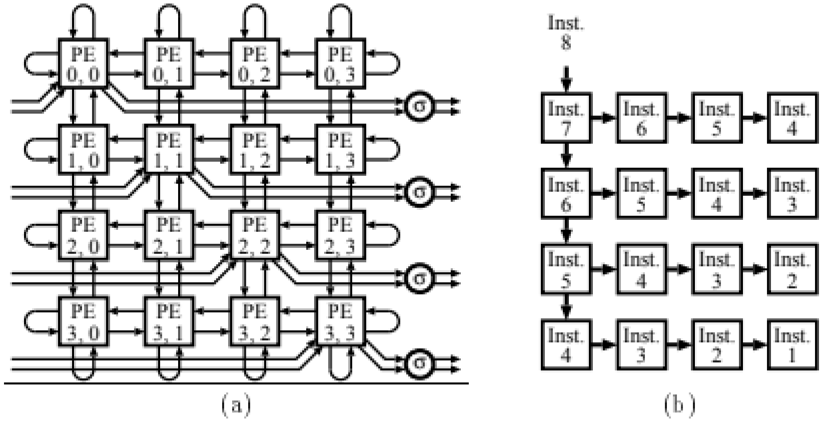
Claim Limitation (Claim 7)	Exemplary Disclosure
	Hebbian learning rule: during this operation, two vectors are injected into the array, flowing north-to-south and west-to-east respectively. The outer product of these vectors is computed, and each synaptic weight is updated with the corresponding element of the resulting matrix. **Kohonen learning rule**: using the same data flow as the previous one, this operation updates the synaptic weights using the Kohonen learning rule. **Maximum element of a vector**: during this operation, the same vector is injected into the array both from north to south and from west to east. At each PE, both operands are compared. The horizontal one is propagated only when it is larger than the vertical one, it is otherwise replaced by the smallest representable number PS_{min}. The resulting vector contains the largest element of the original vector and PS_{min} everywhere else. **Minimum element of a vector**: this operation is the complement of the previous one, the smallest element of the vector being searched for." Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 1-2.  PE 0, 0 PE 0, 1 PE 0, 2 PE 0, 3 PE 1, 0 PE 1, 1 PE 1, 2 PE 1, 3 PE 2, 0 PE 2, 1 PE 2, 2 PE 2, 3 PE 3, 0 PE 3, 1 PE 3, 2 PE 3, 3 σ Inst. 8 Inst. 7 Inst. 6 Inst. 5 Inst. 4 Inst. 3 Inst. 2 Inst. 1 (a) (b) **Figure 1**: (a) Square systolic array of GENES IV PEs. (b) Instruction flow.

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 1-2. "A GENES IV array is a square mesh of simple *processing elements (PEs).* Each PE is connected by serial lines to its four neighbors as shown in figure 1. All input and output operations are performed by the PEs located on the northwest to southeast diagonal The GENES IV structure implements six different operations grouped in three categories." Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf.
[**156c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	As explained below and in the Responsive Contentions Regarding Non-Infringement and Invalidity ("Responsive Contentions"), it would have been obvious to one of skill in the art based on the disclosures in MANTRA I (alone or in combination with the teachings of Tong, Belanovic / Belanovic and Leeser, Shirazi, Aty, Lee, Sudha, Dockser, and GRAPE-3) that the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. MANTRA I was designed to perform neural network algorithms at the minimum required precision. "Fixed-point arithmetic. GENES IV PEs are designed for fixed-point number representation. In the absence of general analytical results on the required precision (see also Section 4.3), simulations of the application described in Section 5 have been used to determine the precision to be implemented." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model*

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	Systolic Neural-Network Accelerator, Scientific Programming, Vol. 5, 1996, at 52. "Contrary to other neural networks, the Kohonen algorithm with quantized weights and inputs has received little attention so far. Three factors influencing its correct convergence can be put in evidence [17]. Clearly, there is a minimal *number of bits* required to encode the weights, depending on the input distribution and dimension, as well as the number of neurons. Second, the adaptation gain α must decrease slowly enough, or have an initially large value, because otherwise the weight updates get rounded to zero before the algorithm has converged. Finally. the neighborhood function should decrease with the distance from the winner neuron, especially if the input dimension is low. These qualitative results were confirmed by a mathematical analysis based on the Markovian formulation of the algorithm [16], giving the necessary and sufficient conditions for the self-organization of the map in the case where the input and weight spaces are one dimensional. Roughly speaking, the results proven for the continuous case [2] also apply in the quantized case if the number of bits is large enough." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 56. "It should be noticed that because of a combined effect of integer arithmetic, batch implementation, and multiple winners, MANTRA I converges with a slightly higher final error than the sequentially implemented floating-point version of the Kohonen algorithm. On a target error rate of 50% more than the minimum error of the original algorithm, the latter and the MANTRA I implementation need, respectively, 12 x 120 and 16 x 120 iterations to reach the desired error rates. This yields an algorithmic efficiency of 12/16 = 75%. To confirm that the algorithmic efficiency on MANTRA I is high and not very far from unity, it was tested with additional data from other applications. Test runs confirmed that the discrepancies between the MANTRA I version and the original version of the Kohonen algorithm are smaller than the standard deviation of the original algorithm itself. For instance, averaging ten runs for each of several sets of learning parameters in the speech codebook classification problem, MANTRA I actually performed better in approximately 50% of the cases." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 58.

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	"In conclusion, the final rate reached by MANTRA I appears acceptable for the power-system application and the number of iterations required is approximately equivalent to that needed by a traditional floating-point version of the Kohonen algorithm." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 59. "The choice of the arithmetic precision of the hardware is quite delicate. A first, typical choice, motivated by hardware simplicity, is to represent all variables in two's complement fixed point. As for the required number of bits for each element, theoretical analysis usually leads to absolute minimum boundaries that, on themselves, cannot guarantee convergence [13]. In the absence of strong analytical grounds, simulation usually provides the designer with the required information. In [2] and [7] simulations are performed to determine the needs of typical backpropagation applications. These have been completed by simulations of the Kohonen model in the application that prompted the development of the present system." Paolo Ienne & Marc A. Viredaz, *GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator*, Proceedings of the International Conference on Application-Specific Array Processors, Venice, Italy, October 1993, at 352. Reduced precision computations were commonly used in connection with neurocomputers like the MANTRA I. *See, e.g.*: "Another simplification made possible by ANN algorithms depends on the reduced precision required in most calculation and in most models (Asanović and Morgan 1991; Holt and Baker 1991; Holt and Hwang 1993; Thiran et al. 1994). As a consequence, the designer may avoid area-expensive floating-point arithmetic units and use reduced integer precision (all but one system among those cited in the following sections use fixed-point arithmetic). This results in arithmetic units, registers and data paths using less area on the die and in a reduction of the physical resources devoted to communication (e.g., less IC pins)." Paolo Ienne Lopez, *Programmable VLSI Systolic Processors for Neural Network and Matrix Computations* (unpublished Ph. D. dissertation, École Polytechnique Fédérale de Lausanne), 1996, at 9-10.

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	"Other systems try to take advantage of other peculiarities of ANN algorithms, such as a reduced precision required in the computations. This makes it possible to develop ad-hoc processing elements which are characterized by a small size and cost. In turn, this often enables one to design systems with a very high processing element count and therefore to increase the degree of parallelism." Paolo Ienne, *Digital Systems for Neural Networks*, PROC. SPIE 10279, Digital Signal Processing Technology: A Critical Review, April 25, 1995, at 11. Accordingly, MANTRA I discloses the use of arithmetic units designed to perform calculations using reduced-precision 16-bit fixed-point math for calculations that typically use 32-bit floating point math, operating on only the 16 most significant bits in the registers. *See, e.g.*: "The registers W_0 and W_1 are 33 bits wide (including an overflow bit), but only the most significant 16 bits are used for non-learning operations. The other registers D, PS, U, and L are all 40 bits wide. Only 16 bits of the register D are however used for other operations than the search for the largest/smallest element of a vector. Similarly, the register PS is used as a 17-bit value for the two learning operations." Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf. "The computation is performed on signed fixed-point values. The inputs and the weights are coded on 16 bits. The weights have 16 additional bits. but these are used only during learning (weight update operations). Outputs are computed on 40 bits." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 50. "BP implementations typically use 32-bit floating point math. This largely eliminates scaling, precision and dynamic range issues. Efficient hardware implementation dictates integer arithmetic units with precision no greater than required. Baker [Bak90] has shown 16-bit integer weights are sufficient for BP training and much lower values adequate for use after training." Hal McCartor, *Back Propagation Implementation on the Adaptive Solutions CNAPS*

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	Neurocomputer Chip, ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 3, 1990, at 1029. For the reasons explained in the Responsive Contentions, it would have been obvious to one of skill in the art to have substituted the fixed-point number format used in MANTRA I for a floating-point format that met the claimed minimum range and precision requirements, and to have used the reduced-precision floating-point number formats disclosed in Tong, Dockser, Belanovic / Belanovic and Leeser, Lee, Shirazi, Sudha, Aty, and TMS 320C32 or the logarithmic format disclosed in GRAPE-3 and Hoefflinger, either alone or in combination. *See also* Appendix to Responsive Contentions (detailing error rates associated with different mantissa sizes).
[**156d**] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit;	MANTRA I discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See, e.g.*: "The MANTRA I machine is controlled by a TMS320C40 digital signal processor (DSP) from Texas Instruments. Two of its six eight-bit built-in communication links connect the machine to another TMS320C40 processor inside a SUN SPARCstation (Fig. 2). From a software point of view, the intermediate DSP is transparent. The MANTRA I machine (the systolic array and its control processor) is completely controlled by the front-end workstation but could be easily integrated into any other computer system based on TMS320C40 processors." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 48. "The structure of the MANTRA I system [18] is shown in Figure 3. The *control module* is the SISD system based on the DSP. It controls the *parallel* or *SIMD module* by dispatching horizontally coded instructions through an FIFO. The SIMD module is frozen when no instruction is pending." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 49.

Exhibit 1 – MANTRA I

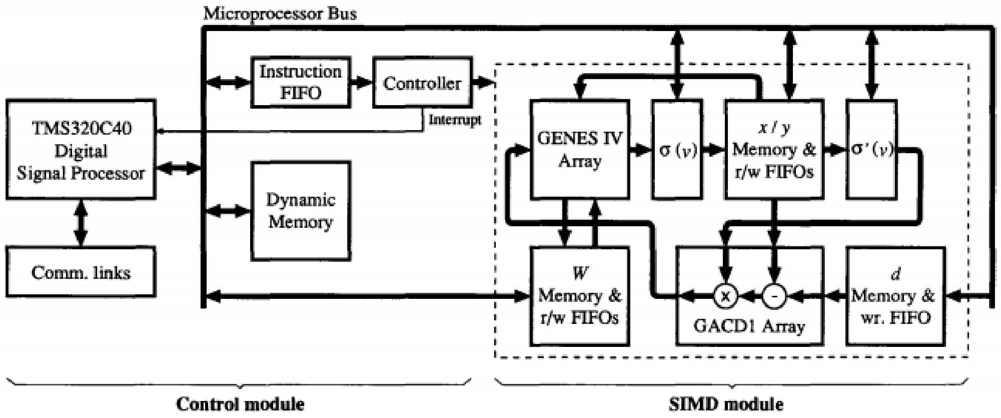
Claim Limitation (Claim 7)	Exemplary Disclosure
	 Microprocessor Bus Instruction FIFO Controller Interrupt TMS320C40 Digital Signal Processor Dynamic Memory Comm. links GENES IV Array σ (v) x / y Memory & r/w FIFOs σ' (v) W Memory & r/w FIFOs x - GACD1 Array d Memory & wr. FIFO Control module SIMD module FIGURE 3 Architecture of the MANTRA I machine. Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 48. “The MANTRA I system [14] is shown in figure 6. The computational heart is a GENES IV array of 40 x 40 PEs. The sequencing of the systolic array is performed by a TMS320C40 digital signal processor (DSP) from Texas Instruments. It also handles the communications with a host workstation and between different MANTRA I machines connected through the dedicated communication links of the DSP.” Paolo Ienne & Marc A. Viredaz, *GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator*, Proceedings of the International Conference on Application-Specific Array Processors, Venice, Italy, October 1993, at 354. “The control part is a complete SISD system based on a commercial microprocessor: the TMS320C40 from Texas Instruments. It configures the SIMD part, dispatches instructions, and manages the inputs and outputs. It also handles the communications with a host computer and between different interconnected MANTRA I computers.” Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 3.

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	"The control part of the MANTRA I machine, groups all the units shown outside the dashed box of figure 4. Its tasks are to configure the SIMD part, to dispatch instructions, and to manage the inputs and outputs. This part is a complete SISD system based on the TMS320C40 microprocessor from Texas Instruments running at 20 MHz. It should also handle the communications with a host computer and between different interconnected MANTRA I computers." Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf.

Exhibit 1 – MANTRA I

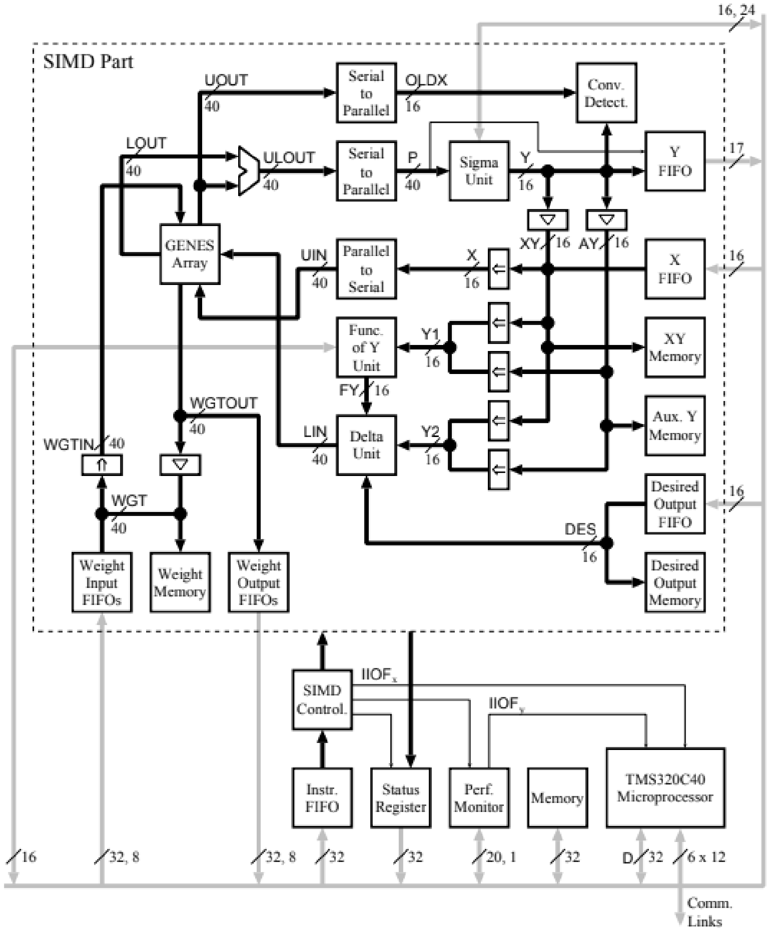
Claim Limitation (Claim 7)	Exemplary Disclosure
	 16, 24 SIMD Part UOUT 40 Serial to Parallel OLDX 16 Conv. Detect. LOUT 40 ULOUT 40 Serial to Parallel P 40 Sigma Unit Y 16 Y FIFO 17 GENES Array XY 16 AY 16 UIN 40 Parallel to Serial X 16 X FIFO 16 Func. of Y Unit Y1 16 XY Memory FY 16 WGTOUT 40 LIN 40 Delta Unit Y2 16 Aux. Y Memory WGTIN 40 WGT 40 Desired Output FIFO 16 DES 16 Weight Input FIFOs Weight Memory Weight Output FIFOs Desired Output Memory IIOF_x SIMD Control. IIOF_y Instr. FIFO Status Register Perf. Monitor Memory TMS320C40 Microprocessor 16 32, 8 32, 8 32 32 20, 1 32 D 32 6 x 12 Comm. Links **Figure 4:** Architecture of the MANTRA I machine. Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf.

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156e**] wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine;	MANTRA I discloses at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine. *See, e.g.*: "The MANTRA I machine is controlled by a TMS320C40 digital signal processor (DSP) from Texas Instruments. Two of its six eight-bit built-in communication links connect the machine to another TMS320C40 processor inside a SUN SPARCstation (Fig. 2). From a software point of view, the intermediate DSP is transparent. The MANTRA I machine (the systolic array and its control processor) is completely controlled by the front-end workstation but could be easily integrated into any other computer system based on TMS320C40 processors." Thierry Cornu, et al. *Design, Implementation, and Test of a Multi-Model Systolic Neural-Network Accelerator*, Scientific Programming, Vol. 5, 1996, at 48. "The MANTRA I system [14] is shown in figure 6. The computational heart is a GENES IV array of 40 x 40 PEs. The sequencing of the systolic array is performed by a TMS320C40 digital signal processor (DSP) from Texas Instruments. It also handles the communications with a host workstation and between different MANTRA I machines connected through the dedicated communication links of the DSP." Paolo Ienne & Marc A. Viredaz, *GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator*, Proceedings of the International Conference on Application-Specific Array Processors, Venice, Italy, October 1993, at 354. "The control part is a complete SISD system based on a commercial microprocessor: the TMS320C40 from Texas Instruments. It configures the SIMD part, dispatches instructions, and manages the inputs and outputs. It also handles the communications with a host computer and between different interconnected MANTRA I computers." Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 3. "The control part of the MANTRA I machine, groups all the units shown outside the dashed box of figure 4. Its tasks are to configure the SIMD part, to dispatch instructions, and to manage the inputs and outputs. This part is a complete SISD system based on the TMS320C40

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	microprocessor from Texas Instruments running at 20 MHz. It should also handle the communications with a host computer and between different interconnected MANTRA I computers." Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf.

Exhibit 1 – MANTRA I

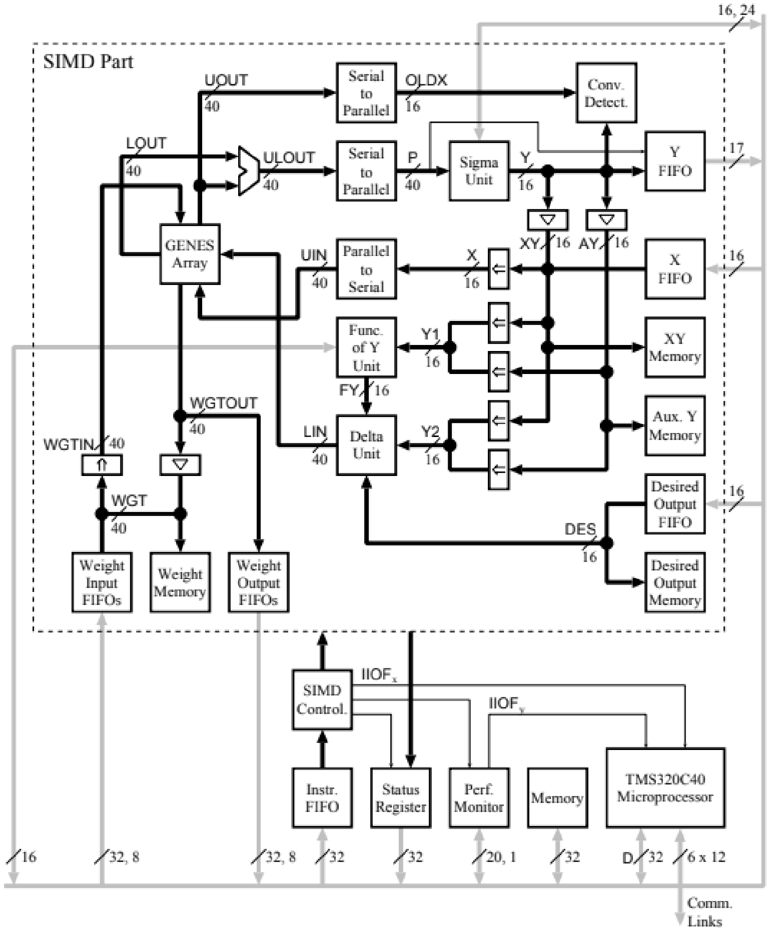
Claim Limitation (Claim 7)	Exemplary Disclosure
	 16, 24 SIMD Part UOUT 40 Serial to Parallel OLDX 16 Conv. Detect. LOUT 40 ULOUT 40 Serial to Parallel P 40 Sigma Unit Y 16 Y FIFO 17 GENES Array UIN 40 Parallel to Serial X 16 XY 16 AY 16 X FIFO 16 Func. of Y Unit Y1 16 XY Memory FY 16 WGTOUT 40 LIN 40 Delta Unit Y2 16 Aux. Y Memory WGTIN 40 WGT 40 Desired Output FIFO 16 DES 16 Weight Input FIFOs Weight Memory Weight Output FIFOs Desired Output Memory IIOF_x IIOF_y SIMD Control Instr. FIFO Status Register Perf. Monitor Memory TMS320C40 Microprocessor 16 32, 8 32, 8 32 32 20, 1 32 D 32 6 x 12 Comm. Links **Figure 4:** Architecture of the MANTRA I machine. Marc A. Viredaz, *MANTRA I: An SIMD Processor Array for Neural Computation,* Spies P.P. (eds) Europäischer Informatik Kongreß Architektur von Rechensystemen Euro-ARCH, 1993, available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9021&rep=rep1&type=pdf.

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156f**] and, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	MANTRA I discloses the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. MANTRA I incorporates 400 to 1,600 processing elements. *See, e.g.*: "The machine, which is currently under test, consists of four printed circuit boards : (1) the processor board (34 chips), (2) the control board (135 chips), (3) the input/output board (465 chips including 14 GACD1 chips), (4) the GENES IV array board (containing a matrix of 10 x 10 = 100 GENES IV chips). One of the latter board is required for configurations of the machine up to 20 x 20 = 400 PEs. For larger configurations—up to 40 x 40 = 1600 PEs—four such boards should be interconnected. A more detailed description of the MANTRA I machine can be found in [vir93]." Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 3. "The heart of the SIMD part in the MANTRA I machine is an array of 40 x 40 GENES IV PEs running at 10 MHz." Marc A. Viredaz & Paolo Ienne, *MANTRA I: A Systolic Neuro-Computer*, In Proceedings of the International Join Conference on Neural Networks, Vol. III, Nagoya, Japan, October 1993, at 3.

Exhibit 1 – MANTRA I

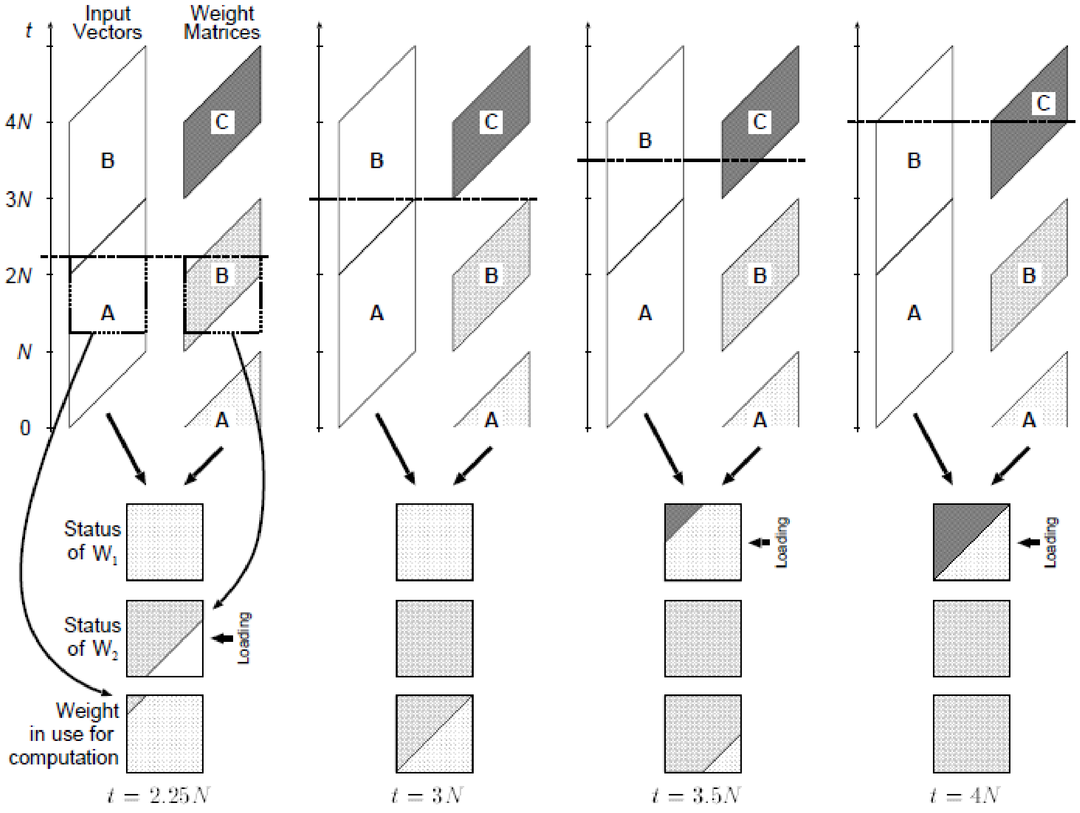
Claim Limitation (Claim 7)	Exemplary Disclosure
	 t Input Vectors Weight Matrices 4N 3N 2N N 0 A B C Status of W_1 Status of W_2 Loading Weight in use for computation $t = 2.25N$ $t = 3N$ $t = 3.5N$ $t = 4N$ Figure 3.11: Genes IV background weight exchange mechanism. The letters **A**, **B**, and **C** identify each weight matrix and the corresponding operand vectors. Paolo Ienne Lopez, *Programmable VLSI Systolic Processors for Neural Network and Matrix Computations* (unpublished Ph. D. dissertation, École Polytechnique Fédérale de Lausanne), 1996, at 44. "The dashed box delimits the parallel or SIMD part of the machine, whose computational heart is an array of up to 40 x 40 Genes IV PEs. The control part is a complete *single instruction-stream single data-stream* (SISD) system based on the TMS320C40 DSP from TEXAS INSTRUMENTS, with its own memory. This processor has six built-in communication channels,

Exhibit 1 – MANTRA I

Claim Limitation (Claim 7)	Exemplary Disclosure
	two of which have been used in the prototype to connect Mantra I to a SUN MICROSYSTEMS workstation where data and programs are stored. The workstation also provides the user interface. Cornu et al. (1996) have described the software upper layer that implements the user front-end." Paolo Ienne Lopez, *Programmable VLSI Systolic Processors for Neural Network and Matrix Computations* (unpublished Ph. D. dissertation, École Polytechnique Fédérale de Lausanne), 1996, at 44.

Exhibit 1 – MANTRA I

’273 Patent

Claim Limitation (Claim 53)	Exemplary Disclosure
[**273a**] A device:	MANTRA I discloses a device. Specifically, the MANTRA I is a neurocomputer. *See* [**156a**].
[**273b**] comprising at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	MANTRA I discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**]
[**273c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X % of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input;	MANTRA I discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**]; *see also* Appendix to Responsive Contentions (detailing error rates associated with different mantissa sizes).

Exhibit 1 – MANTRA I

Claim Limitation (Claim 53)	Exemplary Disclosure
[**273d**] wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	MANTRA I discloses the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide. *See* [**156f**].

Exhibit 1 – MANTRA I

’961 Patent

Claim Limitation (Claim 4)	Exemplary Disclosure
[**961a**] A device comprising:	MANTRA I discloses a device. Specifically, the MANTRA I is a neurocomputer. *See* [**156a**].
[**961b**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	MANTRA I discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	MANTRA I discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**]; *see also* Appendix to Responsive Contentions (detailing error rates associated with different mantissa sizes).
[**961d**] at least one first computing device adapted to control the	MANTRA I discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See* [**156d**].

Exhibit 1 – MANTRA I

Claim Limitation (Claim 4)	Exemplary Disclosure
operation of the at least one first LPHDR execution unit.	

Claim Limitation (Claim 13)	Exemplary Disclosure
[**961e**] A device comprising:	MANTRA I discloses a device. Specifically, the MANTRA I is a neurocomputer. *See* [**156a**].
[**961f**] a plurality of components comprising:	*See* MANTRA I discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* **[156b].** *See also* MANTRA I discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See above* [**156d**].
[**961g**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	MANTRA I discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961h**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first	MANTRA I discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**]; *see also* Appendix to Responsive Contentions (detailing error rates associated with different mantissa sizes).

Exhibit 1 – MANTRA I

Claim Limitation (Claim 13)	Exemplary Disclosure
operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.	